METHOD AND APPARATUS FOR OPERATING A TRANSCEIVER IN DIFFERENT DATA RATES

FIELD OF THE INVENTION

[0001] The present invention relates to data communication, and more specifically to a design for changing the speed of operation of a transceiver.

BACKGROUND OF THE INVENTION

[0002] As a result of improvement in processing technology, it is now possible to put millions of transistors in an integrated circuit. This increases the amount of processing power of the integrated circuit. However, the processing power may be wasted if there are insufficient input-output (I/O) resources to transfer data to and from the integrated circuit. High speed I/O transceivers alleviate this problem by increasing the data transfer speed of the I/O resources.

[0003] A transceiver contains a serializer and a deserializer. A multi-gigabit serializer takes vectors of bits (e.g., 20 bits wide) and converts them to a serial (one bit at a time) bit stream for high-speed transmission over a single differential pair connection. Conversely, a deserializer receives a serial bit stream on a separate differential pair and converts it into vectors of bits. The two circuits together are called a SERDES (serializer/deserializer).

[0004] The rate at which bits are transmitted or received on the differential pair connection is called the serial baud rate, commonly measured in Gbps (gigabits per second), or perhaps Mbps (megabits per second). A number of communication standards exist involving the use of SERDES transceivers. The serial baud rates for some common standards of interest range from 1.06 Gbps to 3.125 Gbps. Other potential applications may use lower serial baud rates,

e.g., 622 Mbps or lower. It is desirable for a single transceiver to be able to operate over this entire range of serial baud rates, especially when it is embedded in a programmable logic device such as a field-programmable gate array (FPGA), which is intended to be used for a broad range of applications.

[0005] It is difficult for conventional circuits like the SERDES to operate at high speed. Thus, many new circuits are specially designed to handle the challenge. Further, the values of components need to be fine-tuned and carefully selected because the SERDES must meet exacting specifications over its entire range of serial baud rates. One result is that a transceiver can optimally operate in a narrow speed range. As an example, some SERDES are designed to operate in the range 800 Mbps to 3.125 Gbps. Such a SERDES is not usable as-is for designs with serial baud rates below 800 Mbps, for example, the 622 Mbps applications noted above. Consequently, it is desirable to be able to extend the range of operation of a SERDES.

SUMMARY OF THE PRESENT INVENTION

[0006] The present invention is a transceiver that contains a serializer designed to operate at a first data rate and a transmitter interface that can optionally lower the data rate of the transceiver. The interface receives a first set of data from a data source at a second data rate and delivers a second set of data to the serializer at the first data rate. The transceiver can also include a deserializer designed to operate at the first data rate and a receiver interface. The receiver interface receives a third set of data from the deserializer at the first data rate and delivers a fourth set of data to a data receiver at the second data rate. A control input to the transmitter and receiver interfaces determines whether the first and second data rates are equal, or whether the second data rate is lower than the first.

[0007] In one embodiment of the present invention, the transmitter interface derives a half-speed clock from a reference clock and optionally provides it to the data source. Using the half-speed clock, the data source supplies data to be transmitted at half the normal rate with respect to the reference clock. The transmitter interface replicates every bit of the data, thereby presenting data of regular lengths to the serializer. As a result, the real data rate is reduced. The opposite operation is used by the receiver interface.

[0008] The above summary of the invention is not intended to describe each disclosed embodiment of the present invention. The figures and detailed description that follow provide additional exemplary embodiments and aspects of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Fig. 1 is a schematic diagram of an integrated circuit containing a plurality of transceivers of the present invention.

[0010] Fig. 2A is a block diagram showing a serializer and a transmitter interface of the present invention.

[0011] Fig. 2B is a block diagram showing a deserializer and a receiver interface of the present invention.

[0012] Fig. 3 shows waveforms of the transmitter interface of the present invention when it operates under one half data speed.

[0013] Fig. 4 shows waveforms of the receiver interface of the present invention when it operates under one half data speed.

[0014] Fig. 5A is a schematic diagram of a portion of a transmitter interface of the present invention.

[0015] Fig. 5B is a schematic diagram of a portion of a receiver interface of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0016] The invention is an interface to the SERDES that cuts the lowest operating serial baud rate in at least half without modifying the SERDES design. Without this technique, it is necessary to modify the SERDES so it can operate at lower serial baud rates while still preserving superior performance at the higher speeds. This is a difficult design problem. For example, the SERDES uses PLLs, which are harder to design for reliable operation over larger frequency ranges.

[0017] A schematic diagram showing an integrated circuit 100 of the present invention is shown in Fig. 1. It contains a plurality of transceivers, such as transceiver 102-104, positioned outside of programmable fabric 106. In one embodiment, programmable fabric 106 comprises field programmable gate array (FPGA) fabric. One or more processor cores, such as processor core 110, can be optionally embedded inside programmable fabric 106. An interface layer 112 is used to facilitate communication between embedded processor core 110 and fabric 106. An example of an interface layer is disclosed in a copending patent application entitled "Programmable Gate Array Having Interconnecting Logic To Support Embedded Fixed Logic Circuitry" (serial number 09/968,446 and filed September 28, 2001). Processor core 110 can be connected to one or more transceivers. In Fig. 1, a pair of routed paths 114-115 are used to schematically show the connection from transceiver 102 to processor core 110 through interface layer 112 and programmable fabric 106. Processor core 110 is preferably designed to process data in high speed. Thus, transceivers 102-104 are preferably high speed transceivers.

[0018] Processor core 110 can be one of a variety of signal processing devices, such as a microprocessor, network processor, video processor. Note that more than one processor core can be embedded inside programmable fabric 106.

[0019] A user may configures a plurality of circuits in programmable fabric 106. Some of these circuits may communicate with transceivers 102-104. For example, Fig. 1 shows a circuit 118 in programmable fabric 106 communicating with transceiver 103.

[0020] Fig. 2A is a block diagram showing a serializer 132 and a transmitter interface 134. In Fig. 2A, transmitter interface 134 denotes that portion of the transceiver that presents data on a bus 136 to serializer 132. In one example, the data is 20 bits wide. Transmitter interface 134 accepts data on a bus 138 originated from a data source 140 within the transceiver, a reference clock (shown as REFCLK 142) that is shared with serializer 132, and a signal 146 for selecting the speed of operation of the transceiver. Transmitter interface 134 generates an interface (i/f) clock 144 so that data in data source 140 can be properly stored into transmitter interface 134.

[0021] Similarly, fig. 2B is a block diagram showing a deserializer 152 and a receiver interface 154. In Fig. 2B, receiver interface 154 denotes that portion of the transceiver that receives data on a bus 156 from deserializer 152. The data is then passed on to a data receiver 160 within the transceiver on a bus 158. Receiver interface 154 accepts a recovered clock (shown as REC_CLK 162) from deserializer 152 and a signal 166 for selecting the speed of operation of the transceiver. Receiver interface 154 also delivers data 158 to data receiver 160 and generates an i/f clock 164 so that data can be properly stored into data receiver 160.

[0022] The SERDES consists of the serializer 132 (Fig. 2A) and deserializer 152 (Fig. 2B) combined.

[0023] In normal full-speed operation, the serializer uses a reference clock whose frequency is 1/20 of the serial baud rate (for a 20-bit vector), for example, 50 MHz for a serial baud rate of 1 Gbps. In one design, the serializer internally stores the 20 bits to be transmitted on the rising edge of the reference clock. Referring to Fig. 2A, the

transmitter interface 134 accepts REFCLK 142 and passes it unmodified to the i/f clock 144 for use by the data source 140. The data source 140 uses the rising edge of the i/f clock 144 to store the data 138 that it presents to the transmitter interface 134. The transmitter interface 134 copies the data 138 unmodified to the data bus 136 connected to the serializer 132. In this usage (full-speed operation), the transmitter circuit behaves as if the transmit interface 134 were not present, with REFCLK 142 directly connected to i/f clock 144, and data 138 directly connected to data 136.

[0024] To reduce the minimum transmission serial baud rate, the transmitter interface also supports an option for half-speed operation. In this mode, the transmitter interface 134 derives from REFCLK 142 a half-speed clock using a negative edge triggered flip-flop, and it presents the half-speed clock on the i/f clock 144 for use by the data source 140. The data source 140 stores a new value on data 138 at each rising edge of i/f clock 144, that is, at every second falling edge of REFCLK 142. For each cycle of REFCLK, the transmitter interface 134 stretches 10 bits of data 138 to 20 bits for presentation on data 136 to the serializer 132. This is done using the half-speed clock as a multiplexer selector input choosing the 10 bits of data 138 to be transmitted. One circuit implementation of this aspect of the invention is shown in Fig. 5A. Each rising edge of REFCLK stores a different set of 10 bits (stretched to 20) into the serializer for transmission.

[0025] As an example, consider this sequence of two 20-bit vectors to be transmitted.

```
        00111110101100001100; and
        00111101000011111010
```

On each rising edge of REFCLK, the transmitter presents 10 bits of the above sequence, stretched to 20 bits by replicating each bit.

```
        0011111010 --> 00 00 11 11 11 11 11 00 11 00
        1100001100 --> 11 11 00 00 00 00 11 11 00 00
        0011110100 --> 00 00 11 11 11 11 00 11 00 00
```

0011111010 --> 00 00 11 11 11 11 11 00 11 00

[0026] Fig. 3 shows the resulting clocking and data timing diagram for half-speed operation of the present invention. It shows a reference clock (REFCLK) 192 that is fed to the serializer. It also shows the i/f clock 194 that is generated by the transmitter interface. It can be seen that the frequency of the interface clock is basically half of REFCLK. Arrows (such as an arrow 196) are used to show the times when data is stored by the data source 140 using i/f clock 194 and by the serializer using reference clock 192. Fig. 3 shows an example of the data vectors 202 and 204 presented on the data connection 138 to the transmitter interface. Each data value 202 and 204 occupies one clock cycle of the i/f clock 194. The corresponding values of data delivered to the serializer are shown as data 206-209. Each data value occupies one clock cycle of REFCLK 192. In this way, the serializer transmits serial data at an effective rate of ten times the REFCLK frequency rather than the normal twenty.

[0027] The inverse operation is done on the receiver side. Referring to Fig. 2B, the deserializer 152 presents 20 bits of parallel data 156 plus a recovered clock 162 to the receiver interface 154. In one design, the 20-bit data changes on falling recovered clock. For normal full-speed operation, the receiver interface 154 passes these data and clock unmodified via data 158 and i/f clock 164 to the data receiver 160. The data receiver 160 uses the rising edge of the i/f clock 164 (that is, the edge during which data 158 is stable) to store data 158.

[0028] To reduce the minimum receiver serial baud rate, the receiver interface also supports an option for half-speed operation. In this mode, the receiver interface 154 derives from REC_CLK 162 a half-speed clock that changes on rising REC_CLK. The receiver interface presents the half-speed clock on the i/f clock 164 for use by the data receiver. On each rising edge of REC_CLK 162, the receiver interface 154 also internally stores alternating bits of data 156, that is, half the bits (10 bits, assuming data 156 is 20 bits wide).

On every second rising edge of REC_CLK 162, corresponding to a falling edge of i/f clock 164, the receiver interface presents on data 158 the 10 bits stored by the preceding rising edge of REC_CLK and the 10 bits stored by the current rising edge of REC_CLK, or 20 bits in all. In this way, the receiver effectively reads 10 bits per REC_CLK cycle, half the normal serial baud rate of the deserializer. One circuit implementation of this aspect of the invention is shown in Fig. 5B.

[0029] Fig. 4 shows various clock waveforms and data for half-speed operation of the receiver interface. It shows a recovered clock waveform 292 generated by the deserializer. Four sets of data, 294-297, are generated by the deserializer, one for each recovered clock cycle. Fig. 4 also shows the half-speed recovered clock signal 304 derived from the recovered clock by the receiver interface. Data vectors 308 and 309 show the 10-bit values stored internally by the receiver interface on the rising edges of the recovered clock 292 corresponding to rising half-speed clock 304. Vectors 308 and 309 consist of alternating bits of vectors 294 and 296, respectively. The half-speed data delivered by the receiver interface to the data receiver is shown as data vectors 312 and 313. The receiver interface stores vectors 312 and 313 on the rising edges of the recovered clock 292 corresponding to falling half-speed clock 304. Vector 312 consists of the previously stored vector 308 joined with alternating bits of vector 295; similarly, vector 313 is composed from vectors 309 and 297.

[0030] An exemplary circuit 250 that can carry out the necessary clock division and multiplexing in a transmitter interface is shown in Fig. 5A. A similar circuit 270 for the receiver interface is shown in Fig. 5B. Circuit 250 comprises three multiplexers 252-254, a flip flop 258 and an inverter 256. The signals in Fig. 5A have the following significance:

full_speed — select speed (corresponding to signal 146 of Fig. 2A):
== 1 for full-speed operation; and

	== 0 for half-speed
REFCLK	SERDES reference clock (corresponding to signals 142 of Fig. 2A and 192 of Fig. 3)
tx_xface_clk_hs	half-speed clock in transmitter interface
tx_xface_clk	transmitter interface clock output (corresponding to signals 144 of Fig. 2A and 194 of Fig. 3)
tx_data_in	transmitter interface data input (corresponding to signals 138 of Fig. 2A and 202-204 of Fig. 3)
tx_data_out	transmitter interface data output (corresponding to data 136 of Fig. 2A and 206-209 of Fig. 3)

[0031] In this embodiment, the rising edge of output clock tx_xface_clk can be used to store tx_data_in at data source 140 of Fig. 2A.

[0032] Circuit 270 comprises two multiplexers, two inverters, two flip-flops, and two registers 272 and 274. Blocks in Figs. 5A and 5B that have the same shape indicate the same types of components. The signals in Fig. 5B have the following significance:

full_speed	select speed signal (corresponding to signal 166 of Fig. 2B): == 1 for full-speed operation; and == 0 for half-speed
rec_clk_in	recovered clock from deserializer (corresponding to signals 162 of Fig. 2B and 292 of Fig. 4)
rx_data_in	receiver interface data input (corresponding to data 156 of Fig. 2B and 294-297 of Fig. 4)
data_ctrl_hs	receiver interface control
rec_clk_out_hs	half-speed recovered clock from receiver interface

data_hold_hs	receiver interface internal register, see data 308-309 of Fig. 4
rx_data_out_hs	half-speed data in receiver interface
rec_clk_out	receiver interface clock output (corresponding to signals 164 of Fig. 2B and 304 of Fig. 4)
rx_data_out	receiver interface data output (corresponding to data 158 of Fig. 2B and 312-313 of Fig. 4)

[0033] One of the many parameters that can characterize the operation of a deserializer is the "run length," i.e., the maximum number of consecutive ones or consecutive zeroes that the deserializer can receive in the incoming serial bit stream while still operating correctly. One aspect of the present invention is that when half-speed operation is selected, the run length is half the value of the run length for full-speed operation, e.g., measured in tenths of REFCLK period instead of 1/20's.

[0034] Another aspect of the present invention is that the receiver circuit is designed to strobe the input serial data as closely as possible to the middle of the bit period. For half-speed, the effective bit period is two SERDES bit periods, so the receiver will not be strobing the input serial data at the middle of the effective bit period, but rather near the 1/4 or 3/4 point. The quality of the lower-speed medium should be able to support this offset in the strobing time.

[0035] It should be noted that other circuits for clock division and alignment with respect to data could be designed. This would be the case when the circuit needs to meet other timing requirements.

[0036] It should also be noted that the principle of the present invention can be applied to reduce the speed of a SERDES by a factor of four or more.

[0037] Those having skill in the relevant arts of the invention will now perceive various modifications and additions which may be made as a result of the disclosure herein. Accordingly, all such modifications and additions are deemed to be within the scope of the invention, which is to be limited only by the appended claims and their equivalents.